

KETTERÄSTI OHJELMISTOKEHITYKSEEN



Ammattikorkeakoulututkinnon opinnäytetyö

Tietotekniikan koulutusohjelma

Riihimäki, Syksy 2017

Mariina Mäkinen

Tietotekniikan koulutusohjelma
Riihimäki

Tekijä	Mariina Mäkinen	Vuosi 2017
Työn nimi	Ketterästi ohjelmistokehitykseen	
Työn ohjaaja/t	Jari Mustajärvi	

TIIVISTELMÄ

Tämän opinnäytetyön tarkoituksena on käydä läpi ohjelmistoprojektimalleja ja ketteriä menetelmiä paneutuen laajemmin Scrumiin sekä Kanbaniin.

Opinnäytteen alussa on teoriaosuus, jossa kerrotaan yleisimmät ohjelmistoprojektimallit sekä ketterät menetelmät. Seuraava luku käsittelee kansainvälistä yhteistyötä, jonka jälkeen tulee tutkimus. Työn lopussa ovat johtopäätökset ja pohdinta, joissa mietitään tutkimuksen tuloksia sekä sitä, miten tutkimus onnistui.

Tutkimuksessa selvitetään mitä ketteriä menetelmiä yritykset käyttävät, miten niitä sovelletaan, mitä ongelmia on kohdattu sekä miten niitä on ratkottu.

Avainsanat Ohjelmistotuotanto, Ohjelmistokehitys, Ketterät menetelmät, Scrum, Kanban

Sivut 20 sivua, joista liitteitä 1 sivu

Information Technology
Riihimäki

Author	Mariina Mäkinen	Year 2017
Subject	Going agile in software development	
Supervisors	Jari Mustajärvi	

ABSTRACT

The aim of this thesis was to explore software project models and agile methods focusing on Scrum and Kanban.

At the beginning of this thesis there is the theoretical part, which explains the most common software project models and agile methods. The next chapter deals with international cooperation, which is followed by research. At the end of this thesis there are the conclusions and the discussion part where the results and success of this research project are examined.

This paper examines the agile methods used in companies and studies how they are as well as lists the problems encountered there and how they have been resolved.

Keywords Software development, Agile methods, Scrum, Kanban

Pages 20 pages including appendices 1 page

SISÄLLYS

1	JOHDANTO.....	1
2	OHJELMISTOPROJEKTIMALLIT	1
2.1	Osa-alueet	1
2.2	Vesiputousmalli	2
2.3	Protoilu	3
2.4	Iteratiivisuus	3
3	AGILE OHJELMISTOTUOTANTO	4
3.1	Yleisesti.....	4
3.2	Agile tiimit	5
3.3	Agile-menetelmät.....	5
4	KANBAN	7
4.1	Periaate	7
5	SCRUM	9
5.1	Periaate	9
5.2	Scrumin roolit	9
5.3	Pyrähdys	10
6	KANSAINVÄLINEN YHTEISTYÖ	12
7	TUTKIMUSMENETELMÄT.....	12
7.1	Kvalitatiivinen menetelmäsuuntaus.....	12
7.2	Tutkimusprosessi.....	12
7.3	Tutkimusongelman määrittäminen ja tutkimussuunnitelma	13
7.4	Teemahaastattelu	13
8	TUTKIMUKSEN SUORITTAMINEN	14
8.1	Yrityksien valinta	14
8.2	Haastattelu	14
8.3	Tutkimustulokset.....	14
9	JOHTOPÄÄTÖKSET	16
10	POHDINTA.....	17
	LÄHTEET	18

Liitteet

Liite 1 Haastattelukysymyksiä

1 JOHDANTO

Tämän opinnäytetyön tarkoituksena on käydä läpi millaisia ketteriä menetelmiä ohjelmistotuotannossa käytetään, paneutumalla eritoten Scrumiin ja Kanbaniin, jotka ovat osoittautuneet suosituimmiksi menetelmiksi haastatelluissa yrityksissä. Ketterät menetelmät ovat tärkeä työkalu ohjelmistoalan yrityksissä, niin projektinhallinnallisesti että riskien minimoimisen kannalta. Haastattelu tapahtui neljässä erilaisessa yrityksessä Helsingissä ja tarkoituksena oli saada samanlaisien haastatteluiden tulokset Brasilialaisista yrityksistä vertailuun. Luvussa 6 selvennän yhteistyön ongelmia ja luku 7 käsittelee itse tutkimusta. Opinnäytetyön lopussa on vielä lukunsa johtopäätöksille sekä pohdinnalle. Tutkimuksen tavoitteena on selvittää mitä menetelmiä yrityksissä käytetään, miten niitä sovelletaan sekä millaisia ongelmia on kohdattu ja miten niitä ratkotaan, tarkoituksena oli verrata menetelmien käyttöä Suomen ja Brasilian välillä ja pohtia minkä takia tuloksissa on eroavaisuuksia. Tietopohjaa opinnäytetyöni teoriaosuuteen hain erilaisista lähteistä. Painettuja lähteitä oli muutamia, mutta erilaisilta verkkosivustoilta sai hyvinkin ajankohtaista tietoa aiheestani.

2 OHJELMISTOPROJEKTIMALLIT

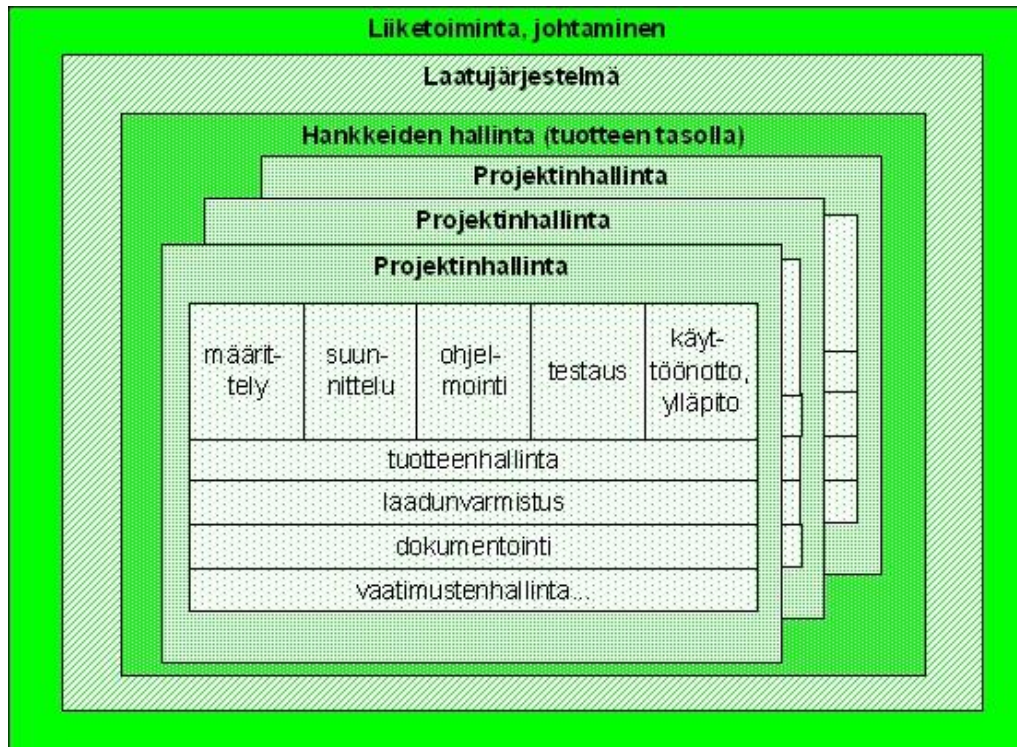
Ohjelmiston tuottamista varten on olemassa lukuisia erilaisia lähestymistapoja, joista yksinkertaisin on niin sanottu ”code-and-hack”. Se on yksinkertaisesti ohjelman kasvattamista, laajentamista, korjaamista ja muuttamista niin kauan, kunnes asiakas on siihen tyytyväinen. Vähänkin monimutkaisempaan ohjelmistoprojektiin toki kuuluu paljon muutakin kuin ohjelmointi. Eri projektimallit eroavat toisistaan ainakin siten, miten eri projektin osa-alueita sovelletaan projektin eri vaiheissa. (Haikala & Mikkonen 2011, 29.)

2.1 Osa-alueet

Ohjelmistoprojekteihin liittyy normaalisti ainakin suunnittelua, määrittelyä, ohjelmointia ja testausta sekä ohjelmiston käyttöönottoon ja ylläpitoon liittyviä asioita. Alla olevassa kuvassa on esitetty ohjelmistotuotannon osa-alueita.

Suunnittelu-termi tarkoittaa teknistä suunnittelua määrittelyn mukaisesta järjestelmästä. Määrittely, eli speksaus merkitsee yleensä projektin asiakasvaatimusten ja niistä tulevien ohjelmiston toimintojen dokumentointia asiakkaan näkökulmasta. Testauksella etsitään virheitä ja se mittaa myös ohjelmiston laatua. Tässä käytössä on yleisesti todentamismenettely nimeltään Faganin vuonna 1976 kehittämä tarkastusmenettely, jossa tarkastetaan ohjelmakoodia tai dokumenttia vertaamalla sitä spesifikaatioon Käyttöönotto tapahtuu ohjelmiston valmistuttua ja siitä alkaa myös ohjelman käyttöiän ylläpitovaihe. Ohjelmis-

totuotteen tilasta pitää kirjata tuotteenhallinta, ja se sisältää ikään kuin sisällysluettelon: mitä komponentteja ja missä versioissa niitä on olemassa. Ohjelman virheitä vähennetään laadunhallinnalla, jota kutsutaan myös nimellä laadunvarmistus. Tällä tarkoitetaan siis menetelmiä, joilla virheitä korjataan. (Haikala & Mikkonen 2011, 30–31 & 1.)

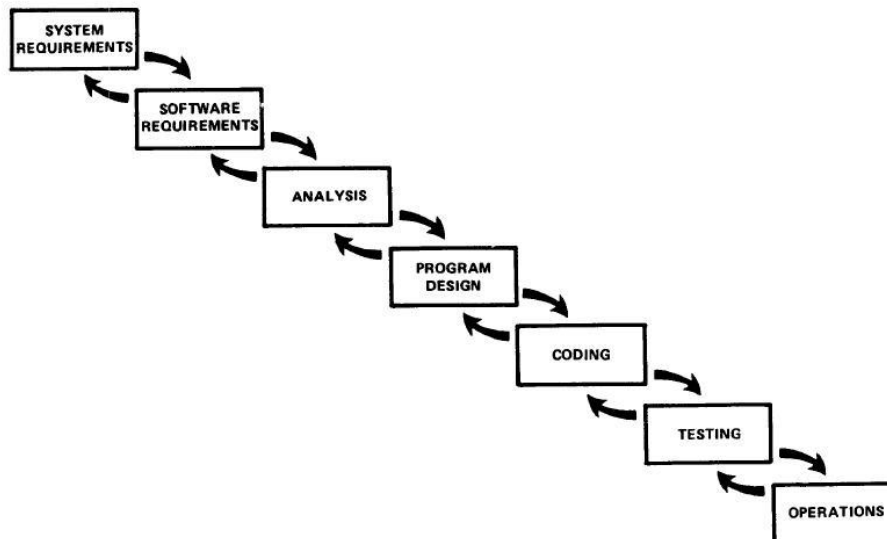


Kuva 1. Ohjelmistotuotannon osa-alueita (Vaara, 2005).

2.2 Vesiputousmalli

Royce julkaisi vuonna 1970 artikkelin: *Managing the Development of Large Software Systems* (Royce 1970), joka on ehkä alan viitatuimmista, mutta myös väärinymmärretyimmistä artikkeleista. Sen merkittävin sisältö on vesiputousmalli. Royce katsoo iteroinnin taaksepäin olevan mallin tärkeä osa, ja hän toteaa, että jos samanlaista ohjelmaa ei ole ennen toteutettu, se kannattaisi tehdä kahteen kertaan. Alla olevassa kuvassa 2, on esitetty vesiputousmallin alkuperäinen muoto.

Vesiputousmalli on yksinkertaistunut teollisuudessa malliksi, jossa ei ole iterointia mukana. Yhä useammin on saatu kuulla väitteitä, miksi vesiputousmallia ei voi käyttää käytännössä syystä tai toisesta. Toisaalta, jos sallitaan iterointi sekä toisen vaiheen aloittaminen jo ennen edellisen valmistumista, on mahdollista saada hyvinkin toimiva toimintamalli, joka sopii moneen tilanteeseen. Useista uusista toimintamalleista on usein huomattavissa vesiputousmallin alkuperäisiä periaatteita hieman muutellusti toteutettuna. (Haikala & Mikkonen 2011, 36–37.)



Kuva 2. Vesiputousmalli (Royce, W. 1970).

2.3 Protoilu

Ohjelmistotuotannossa menetelmää, jossa rakennetaan jollakin tavalla vaillinaisen prototyyppi, jolla voidaan tutkia ohjelmiston joidenkin osien toimintaa, kutsutaan protoiluksi. Protoilussa on kaksi erilaista vaihtoehtoa: evoluutioprototyyppi sekä poisheitettävä tai kertakäyttöinen prototyyppi. Evoluutioprototyyppiä kehitetään nimensä mukaisesti valmiiksi tuotteeksi ja kertakäyttöistä käytetään vain järjestelmän mallintamiseen, jonka jälkeen se heitetään pois ja itse tuotteen valmistus aloitetaan alusta. Yleensä varsinaisen tuotteen valmistus tehdään aivan eri välineillä. Näistä on tietenkin olemassa välimuotoja, jolloin prototyypin osia pyritään hyödyntämään valmiissa tuotteessa. (Haikala & Mikkonen 2011, 38–39.)

2.4 Iteratiivisuus

Useasta peräkkäisestä iteraatiosta muodostuva iteratiivinen (inkrementaalinen) ohjelmistokehitys on kaikkien ketterien menetelmien pohja. Iteraatiot ovat kehityssyklejä, joiden pituus vaihtelee yhden ja kuuden viikon välillä ja jonka lopputuloksena saadaan julkaisu. Jokainen iteraatio on kuin pieni projekti, se sisältää kaikki ohjelmistotuotannon perustehtävät. Toteutettavat toiminnot tehdään joko riski- tai asiakaslähtöisesti. Riskilähtöiseen kehitystyöhön valitaan vaikeimmat ja riskialteimmat tehtävät, tämä lähestymistapa helpottaa projektia, mutta usein syvällä tehdyt hankalat ratkaisut eivät näy toiminnallisuuden parantumisenä asiakkaalle päällepäin. Asiakaslähtöisessä kehitystyössä valitaan asiakkaan priorisoimat tehtävät iteraatioon mukaan. Näin pidetään asiakas tyytyväisenä, mutta riskinä tässä kuitenkin arkkitehtuurin vääristyminen tai ras-

kaan refaktoroinnin tarve. Hyvä vaatimustenhallinta sekä tuotteenhallinta ovat pakollisia, jotta arkkitehtuuri ja dokumentaatio pysyvät oikeanlaisina. (Taina 2009; Haikala & Mikkonen 2011, 42.)

3 AGILE OHJELMISTOTUOTANTO

Agile Alliance –järjestö on ketterien menetelmien edistämistä ajava taho, jonka ”peruskirjana” on julkaistu Agile Manifesto (ketterän ohjelmistokehityksen julistus). Julistuksen perusta on sen arvoissa, jotka on esitetty seuraavasti: ”Löydämme parempia tapoja tehdä ohjelmistokehitystä, kun teemme sitä itse ja autamme muita siinä. Kokemuksemme perusteella arvostamme:

- Yksilöitä ja kanssakäymistä enemmän kuin menetelmiä ja työkaluja
- Toimivaa ohjelmistoa enemmän kuin kattavaa dokumentaatiota
- Asiakasyhteistyötä enemmän kuin sopimusneuvotteluja
- Vastaamista muutokseen enemmän kuin pitäytymistä suunnitelmassa

Jälkimmäisilläkin asioilla on arvoa, mutta arvostamme ensiksi mainittuja enemmän.”

Manifestin takana on useita periaatteita, joita ketterissä menetelmissä noudatetaan. Tärkein tavoite on tyydyttää asiakas toimittamalla ohjelmiston versioita, jotka täyttävät hänen tarpeensa, säännöllisesti sekä jo aikaisessa vaiheessa. Versioita toimitetaan parin viikon tai kuukauden välein. Projektit rakennetaan motivoituneiden henkilöiden ympärille ja sen parissa työskennellään yhdessä ohjelmistokehittäjien ja liiketoiminnan edustajien kanssa päivittäin. Ohjelmiston oikean rakenteen ja teknisen tason alituinen huomiointi edistää ketteryyttä. Parhaat rakenteet, vaatimukset ja suunnitelmat tulevat itseorganisoiduista tiimeistä, jotka tarkastelevat toistuvasti kuinka parantaa tehokkuuttaan ja mukautuvat sen suhteen. (Agile Manifesto, 2001.)

3.1 Yleisesti

Agile eli ketterä kehitys on yleisnimitys joukolle ohjelmistotuotannossa käytettäviä menetelmiä ja käytäntöjä perustuen Agile Manifeston arvoihin ja periaatteisiin.

Kehittäjille annetaan vastuuta oman työnsä suunnittelusta sekä toteutuksesta, ja tiimin sisällä sovitaan työtehtävistä sekä kehitetään työmenetelmiä jatkuvasti. Työskentely pohjautuu toimivaan kommunikaatioon tiimin sisällä ja sen ulkopuolella. Ketterään kehitykseen kuuluvat oleellisesti työn paloittelu pienempiin osiin ja valmiiden osioiden esittely asiakkaalle tietyin väliajoin. Asiakas onkin mukana projektissa sen koko kehityksen ajan, jolloin saadaan palautetta ja ohjelmistoa voidaan kehittää jatkuvasti oikeaan suuntaan. (Lehtonen, Tuomi-

vaara, Rantala, Käsälä, Mäkilä, Jokela, Könnölä, Kaisti, Suomi, Isomäki & Ylitolva 2014, 2-3.)

3.2 Agile tiimit

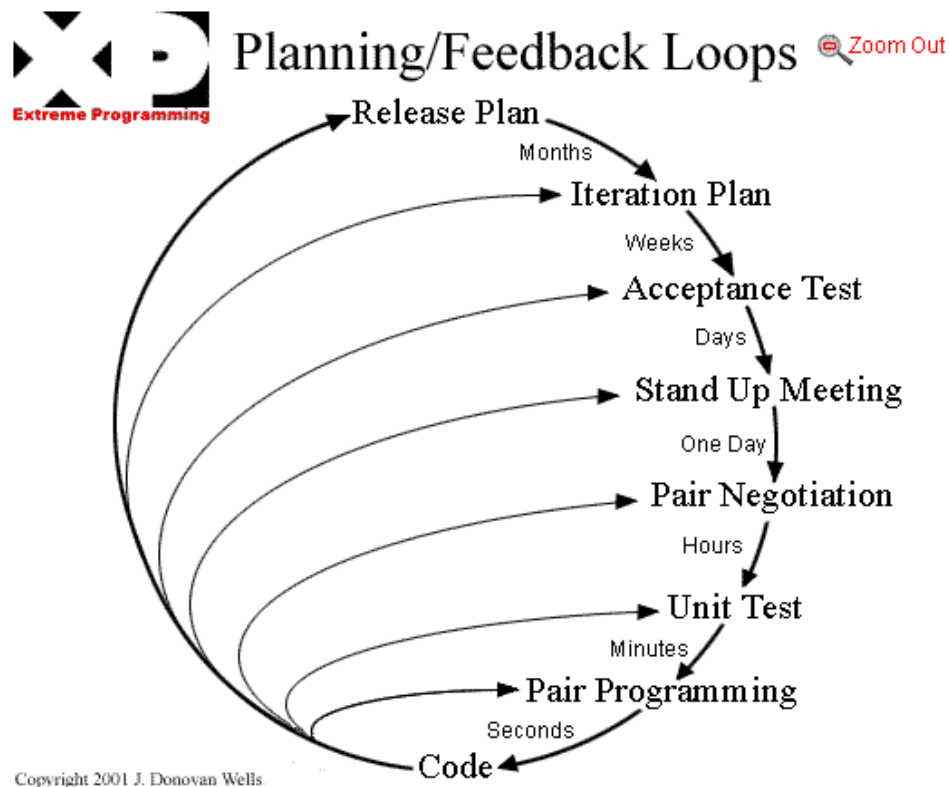
Don Wells kirjoittaa ketterien tiimien tärkeydestä agile-process.org – sivustolla. Hän on sertifioitu Scrum-mestari, joka työskentelee konsulttina, kirjailijana ja luennoitsijana Extreme Programming ja Agile-muutoksien tiimoilta. Wells painottaa, että tiimin tapa työskennellä on tärkeämpää kuin mikään prosessi. Se, että mahdollistaa tiimien tehokasta työskentelyä yhtenäisenä yksikkönä voi parantaa tuottavuutta moninkertaisesti. Ohjelmistokehityksen rahoittavan asiakkaan ottaminen tiimin keskeiseksi ja arvokkaaksi jäseneksi edesauttaa haluttua lopputulosta, sillä asiakas pystyy projektin aikana tekemään pieniä päätöksiä ja muutoksia sen sijaan, että lopuksi katsotaan tuliko projektista haluttu. Asiakkaan työskennellessä kehitystiimin kanssa oppivat kaikki uutta työn alla olevasta projektista tai ongelmasta. Dokumenteissa voi olla runsaasti tietoa, mutta todellista tietoa on vaikea saada paperille, siksi jokapäiväinen työskentely asiakkaan kanssa takaa oikeanlaisen systeemin.

Kehitystiimien tulee olla monialaisia ja itseohjautuvia, tarkoittaen sitä, että tiimin velvollisuuksiin kuuluu päättää miten järjestäydytään ja mitä tehdään seuraavaksi. Vastuu itsensä johtamisesta hyväksytään ja samalla etsitään mahdollisia ongelmia ja lievennetään riskejä. Ihmiset luovat tunnesiteitä projekteihinsa ja ketterissä sellaisissa ongelma on vielä suurempi, koska projektin parissa työskennellään tiiviisti yhdessä. Jotkut ihmiset jopa pelkäävät loukkaantuneensa jos projekti epäonnistuu, ja osa heistä yrittää lähteä ennen sitä. Tämä voi näyttää siltä, että tiimin jäsenet haluavat tahallaan vahingoittaa projektia, johtajat saatavat sitten ihmetellä miksi niin tehdään ja harkita jäsenien korvaamista. Tämä kuitenkin on väärä lähestymistapa. Henkilöiden korvaamisen sijaan tulisi miettiä miksi tiimin jäsenet pelkäävät projektin epäonnistumista ja sitten korjata asiaa projektin puolelta. (Wells 2009.)

3.3 Agile-menetelmät

Extreme Programming (XP) on yksi suosituimmista ketteristä menetelmistä. Se pohjautuu viiteen pääkohtaan: kommunikaatioon, yksinkertaisuuteen, palautteeseen, kunnioitukseen ja rohkeuteen. Ohjelmoijat kommunikoivat keskenään ja asiakkaan kanssa jatkuvasti, pitävät kehitysvaiheessa olevan ohjelmansa siistinä ja yksinkertaisena ja lisäksi testaamalla he alkavat saada palautetta työstänsä heti sen kehityksen alussa. Alla olevassa kuvassa 3 on esitetty XP:n suosittuimmat palautesilmukat tai kanavat. XP:ssä on käytössä lyhyet kehityssykli, jotka tarjoavat syklien välissä olevia tarkastuspisteitä. Tarkastuspisteiden ansiosta ohjelman eri versioita toimitetaan asiakkaalle mahdollisimman varhain, jotta kehitykseen saadaan viipymättä otettua mukaan mahdolliset muutokset sekä vaihtuneet asiakasvaatimukset. Tähän perustuen XP-ohjelmoijat pystyvät vastaamaan muuttuviin vaatimuksiin ja teknologioihin rohkeasti. (Wells 2013;

Lehtonen, Tuomivaara, Rantala, Käsälä, Mäkilä, Jokela, Könölä, Kaisti, Suomi, Isomäki & Ylitolva 2014, 6-7.)



Kuva 3. Suunnittelu/palautesilmukat (Wells D. 2001).

Lean-nimitys on annettu Toyotan menetelmälle tuottaa ja kehittää autoja, sen taustalla olevien periaatteiden ansiosta Lean toimii työelämän monissa tarkoituksissa. Lean on enemmänkin ajattelutapa kuin menetelmä, se pohjautuu hukan eliminointiin sekä ihmiskeskeisyyteen. Hukkaa on kaikenlainen työ, joka ei tuota lisäarvoa asiakkaalle, myös liian aikaisin ns. "varastoon" tehty työ on hukkaa. Ihmiskeskeisyyden tärkeimpänä asiana esillä on kunnioitus johtoportaalle ja työntekijöiden välillä. Kunnioitus tarjoaa projektiin joustavuutta ja jatkuvaa kehitystä sekä pitää ammattitaitoiset henkilöt viehättyneinä palveluksessaan työhönsä. (Shalloway, Beaver & Trott 2009, 5-10; Haikala & Mikkonen 2011, 54-55.)

Dean Leffingwellin kehittämä The Scaled Agile Framework (SAFe) on ikään kuin yhdistelmä useiden ketterien menetelmien periaatteita. SAFe on tarkoitettu suurille ohjelmisto- ja systeemiprojekteille sekä koko organisaation ketteröittämiseen. SAFe on jaettu neljään tasoon: portfoliotaso, arvovirta, hanketaso ja tiimitaso.

Portfoliotasolla ohjeistetaan pitkän aikavälin strategian kokoamista, rahallisten arvovirtojen sekä suurempien aloitteiden hallitsemista. Arvovirta-taso on käytössä vain maailman suurimmissa ohjelmisto- ja systeemiprojekteissa, se sisältää ratkaisujen hallinnan, insinööritieteet ja arkkitehtuurit, ketterät asiakas- ja

toimittajasuhteet sekä arvovirran koordinoinnin. Hanketasolla käytetään sykleittäin eteneviä toteutusjunia (Agile Release Train), tiimeillä sama päämäärä, arkkitehtuurin ja käyttäjäkokemusten hallinta, taso tarjoaa todisteet edistymisestä ja sopivuudesta määrättyyn tarkoitukseen. Tiimitasolla tarkoitetaan itseorganisoituvia ja itseohjautuvia tiimejä, jotka ovat myös monialaisia (cross-functional). Tiimit soveltavat Scrumia, Kanbania ja XP:ia ja toimittavat hyvin testattua ja toimivaa ohjelmistoa ja systeemiversiota aina kahden viikon välein. (Scaled Agile Inc 2016.)

DevOps on ketterä ohjelmistokehitysmalli, joka painottuu ohjelmistokehitys-, ylläpito- ja tuotantohenkilöiden väliseen kommunikaatioon ja näin tehostaa ohjelmistokehitystä. DevOpsissa korostuu ketterien menetelmien nopeat kehitysjaksot ja siinä käytettävä automatisointi nopeuttaa ja parantaa työn laatua. DevOps pohjautuu kolmeen tukipilariin; henkilöt, prosessit ja työkalut. Henkilöpilari käsittää työskentelykulttuurin, jossa kommunikaatio toimii ja roolien välillä ei synny vastakkainasettelua. He tekevät yhteistyötä yhteisen määränpään saavuttamiseksi ja tapaavat päivittäin. Tehokkaan yhteen toimimisen takaamiseksi henkilöiden tulee noudattaa yhteisesti jaettua prosessia. DevOps käsittää koko palvelun, aina ohjelmakoodista testaukseen ja ylläpitoon. Jatkuva julkaisu on suuressa osassa ja sen tärkeimpiä periaatteita ovatkin saada välitöntä palautetta jokaisesta työn vaiheesta. Työkaluja hyödynnetään prosessin tukena eri vaiheiden automatisoinnissa sekä projektinhallinnassa. Niiden tehtävänä on lisätä tuottavuutta automatisoimalla työvaiheita ja helpottaa itse työn tekemistä. (DevOps-tehostettua ohjelmistokehitystä, n.d.)

4 KANBAN

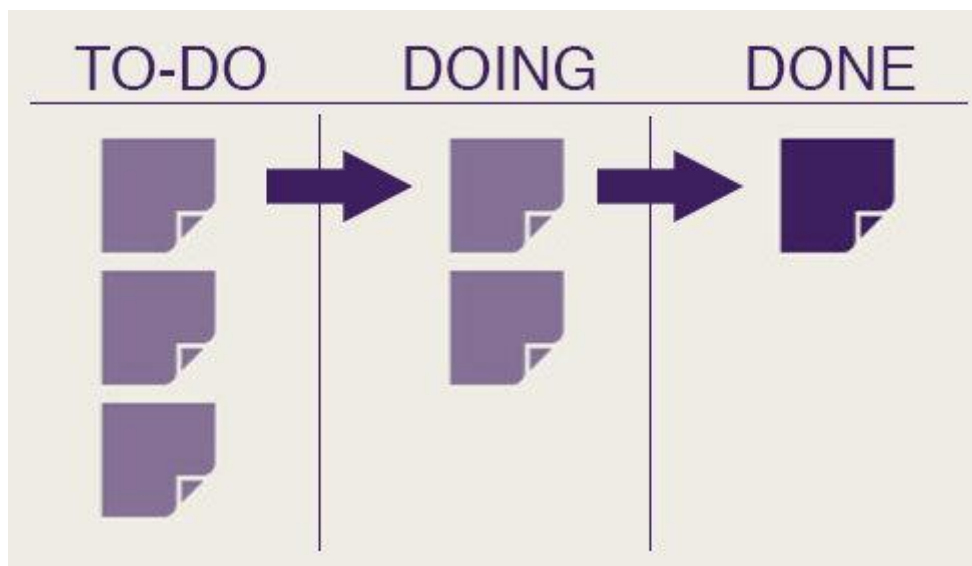
Kanban-sana on japania ja se tarkoittaa näkyvää taulua tai korttia. Toyotan linjatyöntekijät ovat käyttäneet Kanban-kortteja esittääkseen missä vaiheissa heidän valmistusprosessit ovat, korttien visuaalisuuden vuoksi kommunikointi tiimeissä on ollut helppoa. (LeanKit Inc. 2017.)

4.1 Periaate

Kanban on työnhallintatapa, joka perustuu yksinkertaisuudessaan kolmeen periaatteeseen.

1. Työn etenemisen visuaalinen esittäminen, joka tarkoittaa Scrumissa käytetyn tehtävätaulun kaltaisen taulun ylläpitoa. Taulussa tulisi olla erinimiset sarakkeet eri vaiheissa oleville töille, esimerkiksi *ei aloitettu*, *työn alla*, *tarkastuksessa* ja *valmis*. Taulun avulla voidaan helposti seurata töiden etenemistä.
2. Kaistanrajoittimet, joilla rajoitetaan kussakin vaiheessa olevien töiden määrää.
3. Läpimenoajan mittaamisella tarkoitetaan kussakin työn vaiheessa vietettyä aikaa, tätä tarkkailemalla voidaan kehitysprosessia optimoida.

Kanbanin tärkeä tavoite on optimoida arvoketjua, joka tapahtuu juurikin työn etenemisen seurannalla. Arvoketjua tarkastelemalla saadaan selville ongelmakohtia ja niihin pystytään etsimään ratkaisuja. Kanban perustuu siis visuaalisuuteen ja siihen, että ihmisen aivot käsittelevät kuvia helpommin kuin tekstiä. Kanban-taulu luo ikään kuin kuvan meneillään olevasta työstä, jolla pystyy samalla kommunikoimaan tiimin sisällä. Kanbanissa ei ole käytössä pyrhdyksiä, kuten Scrumissa, vaan uusi tehtävä aloitetaan aina kun kaistalla on tilaa. Alla olevassa kuvassa on esitetty esimerkki yksinkertaisesta Kanban – taulusta. (Lehtonen, Tuomivaara, Rantala, Käsälä, Mäkilä, Jokela, Könnölä, Kaisti, Suomi, Isomäki & Ylitolva 2014, 8; Haikala & Mikkonen 2011, 54-55; LeanKit Inc. 2017.)



Kuva 4. Yksinkertainen Kanban – taulu (LeanKit Inc. 2017).

5 SCRUM

Scrum on lähes vakiintunut sanan ketterä synonyymiksi johtuen sen suosiosta ohjelmistosuunnittelijoiden keskuudessa. Se on otettu pääosin hyvin lämpimästi vastaan. Scrumin perusperiaatteet ovat helposti selitettävissä ja Scrumin suosio perustuukin sen yksinkertaisuuteen. Scrumista vaikuttaa löytyvän vastaus hyvinkin moniin ongelmiin ohjelmistotuotannossa. (Haikala & Mikkonen 2011, 46.)

5.1 Periaate

Scrumin suurin etu on sen yksinkertaisuus. Sen perusajatus on selitettävissä nopeasti ja jo ensitarkastelulla se vaikuttaa tarjoavan vastauksia moniin ohjelmistotuotannon ongelmiin. Scrum ei kuitenkaan ota kantaa käytettyihin kehitysmenetelmiin ja työkaluihin, joten karkeasti sanottuna se on projektin toteutusvaiheeseen osoitettu tapa organisoida projektin iteraatiot. (Haikala & Mikkonen 2011, 47.)

Scrum pohjautuu empirismiin, tai empiriseen prosessinhallintateoriaan. Empirismen mukaisesti tieto perustuu kokemukseen sekä päätösten tekemiseen tutujen tosiasioiden pohjalta. Riskien kontrolloimiseen ja ennustettavuuden tehostamiseen Scrum käyttää iteratiivis-inkrementaalista (toistavaa ja lisäävää) toimintatapaa. Kolme tukijalkaa; läpinäkyvyys, tarkastelu ja sopeuttaminen ovat empirisen prosessinhallinnan tärkeimmät osat. (Schwaber & Sutherland 2016, 3.)

5.2 Scrumin roolit

Scrum-mestari on vastuussa siitä, että kaikki ymmärtävät ja käyttävät Scrumia. Häntä voidaan luonnehtia nimellä ”projektipäällikkö ilman valtaa”, Scrum-mestari toimii tiimin palvelevana johtajana. Hän vastaa pyrähdysten tuloksesta ja varmistaa ettei työtä merkata valmiiksi, jos se ei täysin täytä kaikkia määritellyjä ehtoja. Hänen velvollisuuksiin kuuluu myös vastata tiimin hyvinvoinnista ja poistaa haitaksi olevat esteet sekä auttaa tiimin ulkopuolisia ymmärtämään mitkä tavat ovat käyttökelpoisia ja mitkä eivät. Näin Scrum-mestari voi muuttaa tapoja ja maksimoida tiimin tuottaman työn arvoa. (Schwaber & Sutherland 2016, 6; Haikala & Mikkonen 2011, 49.)

Scrum-tiimin ihanteellinen koko on riittävän pieni ollakseen ketterä, mutta silti riittävän suuri, jotta pyrähdysten aikana saadaan riittävä määrä työtä valmiiksi. Optimaalinen koko on yleensä seitsemän kokopäiväistä ja mieluiten eritaustaista työntekijää. Tiimin jäsenillä voi olla erilaisia työn painopisteitä tai osaamista, mutta vastuu tuotteen kehityksestä kuuluu koko tiimille. Tiimi on itseorganisoi-tuva, joten projektipäällikköä ei varsinaisesti ole. Kehitystiimin tulisi työskennellä yhdessä tilassa, jonne voidaan sijoittaa tehtävätaulu, josta voidaan seurata työn edistymistä vaiheittain (esimerkiksi ei aloitettu, työnalla, valmis ja hyväk-

sytty). Taulua ylläpidetään yhdessä tiimin jäsenten sekä scrum mestarin kanssa. Päivittäiset Scrum-kokoukset (daily scrum) voidaan pitää taulun edessä. (Schwaber & Sutherland 2016, 6; Haikala & Mikkonen 2011, 49.)

Tuotteen omistaja on vastuussa projektin taloudellisesta tuloksesta ja toimii ikään kuin yhdyshenkilönä kaikkiin sidosryhmiin, joilta hän kokoaa ohjelmiston vaatimukset tuotteen työlistaksi. Hän myös pitää yllä työlistaa prioriteettien mukaan jäsenneltynä. Työlista sisältää kohtien selkeän ilmaisun sekä niiden järjestämisen niin, että tavoitteet saadaan saavutettua oikein, kehitystiimin työn arvon tarkistamisen sekä hyödyntämisen maksimaalisesti. Myös tuotteen työlistan avoimuuden, läpinäkyvyyden ja ymmärrettävyyden varmistamisen ja sen, että työlistasta selviää mitä Scrum-tiimi tulee tekemään seuraavaksi. Tuotteen omistajan tulee myös varmistaa, että tiimi ymmärtää työlistan asianmukaisella tavalla. (Schwaber & Sutherland 2016, 5; Haikala & Mikkonen 2011, 48.)

5.3 Pyrähdys

Pyrähdys on Scrumin ydin, sen on kestoltaan enintään kuukauden mittainen työjakso. Pyrähdykset alkavat jokaisen pyrähdyksen alussa pyrähdyksen suunnittelupalaverilla, joka kestää kokonaisen päivän ja johon osallistuvat tiimi, tuotteen omistaja ja Scrum-mestari. Palaverin alku käytetään tuotteen omistajan esittelemän työlistan läpikäyntiin ja siihen tutustumiseen. Loppu päivästä päätetään mitä työlistan alkioita otetaan mukaan pyrähdykseen. Tästä syntyvää tehtävää ei saa muuttaa pyrähdyksen aikana. Pyrähdyksen suunnittelussa selviää millaisen version tuotteesta on mahdollista toimittaa ja miten sen toimittamiseen liittyvä työ voitaisiin toteuttaa. Suunnittelupalaverin pohjana on tuotteen työlista, viimeisin tuoteversio ja tiimin aiempi suorituskyky. (Haikala & Mikkonen 2011, 49-51; Schwaber & Sutherland 2016, 7-8.)

Pyrähdyksen tarkkailuun käytetään pyrähdyksen edistymiskäyrää, jota päivitetään jatkuvasti kun tehtäviä tulee tehdyksi. Näin nähdään helposti, miten pyrähdyksen työmäärä kehittyy.

Pyrähdyksen ollessa käynnissä, pidetään joka päivä päivän Scrum-palaveri (daily scrum), jossa jokainen tiimin jäsen kertoo mitä on tehnyt edellisen palaverin jälkeen, mitä aikoo tehdä seuraavaksi ja sen, onko tiedossa esteitä, jotka voivat hidastaa työn tekoa. Päivän Scrum-palaveri on tärkeä osa Scrumia, scrum-mestari on vastuussa sen pitämisestä. Palaveri parantaa kommunikaatiota, karsii muita palavereita, tunnistaa ja poistaa työn hidastimia, kohentaa tiimin asiantuntemusta ja edistää ripeää päätöksentekokykyä. (Haikala & Mikkonen 2011, 50-51; Schwaber & Sutherland 2016, 10.)

Pyrähdyksen katselmointikokous pidetään pyrähdyksen päätyttyä, ja sen kesto on normaalisti noin neljä tuntia. Tiimi demonstroi pyrähdyksen tulokset ja kerrää palautteen sidosryhmien edustajilta. Se keskustelee myös siitä, mitä seuraavaksi lähdetään kehittämään. Scrum-mestari on vastuussa tästäkin kokouksesta ja varmistaa, että se pidetään asetetun aikarajan sisällä sekä sen, että

osallistujat ymmärtävät kokouksen tarkoituksen. (Haikala & Mikkonen 2011, 51; Schwaber & Sutherland 2016, 10-11.)

Pyrähdysten arviointipalaveri (retrospective) pidetään katselmointi- ja suunnittelukokouksien välissä ja sen kesto on enintään kolme tuntia. Arviointipalaverissa arvioidaan kuinka pyrähdys on onnistunut ja mietitään miten toimintatapoja voitaisiin kehittää. (Haikala & Mikkonen 2011, 51; Schwaber & Sutherland 2016, 11-12.)



Kuva 5. Scrum-prosessi (Free Scrum Master Training 2017).

6 KANSAINVÄLINEN YHTEISTYÖ

Yhteistyötä on tehty Brasilialaisesta Feevale University – yliopistossa opiskelevien Camila Utzin ja Matheus Cardoson kanssa. Feevale University on Brasilialainen yliopisto, joka on perustettu vuonna 1969. Se sijaitsee Novo Hamburgo – nimisessä kaupungissa.

Utzin ja Cardoson oli tarkoitus tehdä projekti ketterien menetelmien käytöstä ohjelmistotuotannossa ja suorittaa samanlainen tutkimus. Minusta riippumattomista syistä haastattelut eivät toteutuneet Brasiliassa toivotulla tavalla, enkä saanut heiltä haastattelutuloksia. Toimitin tekemieni haastattelujeni tulokset heille kuitenkin. Tästä syystä tutkimukseni jää vajaaksi, sillä puolet tuloksista jäi saamatta. Etenen kuitenkin omilla tuloksillani ja pyrin parhaaseen mahdolliseen lopputulokseen.

7 TUTKIMUSMENETELMÄT

Tutkimus suoritetaan haastattelututkimuksena, tiedon luonne pohjautuu empirismiin, joka tarkoittaa haastateltavan henkilön kokemusta todellisuudesta. Täytyy pitää mielessä, että ihminen on erehtyväinen ja tieto on enemmän tai vähemmän virheellistä. (Kurkela, n.d.)

7.1 Kvalitatiivinen menetelmäsuuntaus

Kvalitatiivisessa eli laadullisessa tutkimuksessa tiedonkeruumenetelmänä on haastattelu. Kyse on enemmän tai vähemmän avointen kysymysten tai keskusteluteemojen esittämisestä valituille yksilöille tai ryhmille. On tärkeää, ettei tutkija sekoita omia ajatuksia ja asenteitaan tutkimuskohteeseen, vaan yrittää ymmärtää haastateltavan näkökantoja ja ulosantia. Pyritään vuorovaikutukseen haastateltavan kanssa. (Kurkela n.d.)

7.2 Tutkimusprosessi

On tärkeää, että tutkijalla on jo tutkimuksen alussa tarpeeksi selkeä kuva tutkimusprosessista kokonaisuutena. Tutkimussuunnitelman kulminaatiopisteiksi voidaan nimittää kolme vaihetta:

1. tutkimusongelman muotoilu
2. aineiston keruun aloittaminen
3. analyysin aloitus

Näiden vaiheiden merkitys on siinä, ettei uutta vaihetta voi aloittaa ennekuin edellinen on valmis. Tutkimussuunnitelmaan vaikuttaa myös kolme muuten oleellista asiaa, jotka tulee olla näkyvillä alusta asti:

- aikataulu

- raha ja muut aineelliset resurssit
- henkiset yms. resurssit

Täytyy pitää mielessä, että tutkimusongelman määrittelyyn ja tutkimuksen suunnitteluun voi mennä jopa puolet koko tutkimuksen ajasta. (Kurkela n.d.)

7.3 Tutkimusongelman määrittäminen ja tutkimussuunnitelma

Yleensä tutkimuksen ensisijainen tavoite on saada lisää tietoa jostakin ongelmasta, tämä ongelma on niin sanottu asiaongelma, joka harvoin on suoraan tutkimusongelma. Asiaongelmaa joudutaan usein määrittämään ja täsmentämään, jotta siitä saadaan tutkimusongelma. Itse tutkimus paneutuu siis tutkimusongelmaan, mutta loppuraportissa siirrytään takaisin pohtimaan mitä tulokset merkitsevät alkuperäisen asiaongelman kannalta.

Määriteltäessä asiaongelmaa pohditaan mitä tekijöitä analysoitavaan ilmiöön liittyy. Tekijät priorisoidaan olennaisuuden mukaan sekä otetaan selvää millaista tietoa ilmiöstä on jo olemassa.

Tutkimuksen merkittävä osa on tutkimusongelman määrittäminen. Määrittely tapahtuu tutkimussuunnitelman kirjoittamisella. Tutkimussuunnitelmassa kuvataan teoreettinen näkökulma, viitekehys sekä keskeiset käsitteet. Tarvittavat tiedot yksilöidään, tutkimusote ja tutkimusmenetelmä valitaan sekä tutkimusasetelma määritellään. Suunnitelmassa päätetään myös otoksesta ja aineiston käsittelymenetelmistä sekä tehdään raportointisuunnitelma, kustannusarvio ja aikataulu. (Kurkela n.d.)

7.4 Teemahaastattelu

Teemahaastattelussa on erityisen tärkeää, että esiin nostettavat aiheet ovat tarkoin pohdittuja ja määriteltyjä. Teemahaastattelu on siis keskustelua, jolla on etukäteen tarkoin valittu tarkoitus ja jota haastattelija ohjaa hallitusti. Etuna teemahaastattelussa on se, että saatava tieto koostuu aidosti haastateltavan henkilön kokemuksista. On oltava tarkkana, ettei haastattelija lähde liikaa johdattamaan haastattelun kulkua.

Laadulliseen tekstiin ja teemahaastatteluun pohjautuvan materiaalin tulkinnessa on kaksi etenemistietä.

- Toinen on aineistossa tiukasti pitäytyminen. Tulkitaan ja tarkastellaan vain aineistossa olevia seikkoja.
- Toisessa aineistoa pidetään apuvälineen tehtävänä, silloin se on lähtökohtana tulkinnoille ja teoreettiselle pohdinnalle.

Tulkintojen tekemiseen ei oikeastaan ole oikeita ohjeita olemassa, se saattaakin olla haastavin osuus laadullisessa tulkinnessa. Tutkijan täytyy luoda yleisemmän tason tulkintoja haastateltavien kertomista, niin sanotuista arkielämän tilanteista, joita voidaan pitää ensimmäisen asteen tulkintoina. Tutkijan tulee pyrkiä

teoreettiselle ajatuksen tasolle, jolloin haetaan yleisemmällä tasolla päteviä teoreettisia tulkintoja. (Kurkela n.d.)

8 TUTKIMUKSEN SUORITTAMINEN

Tutkimuksen päämääränä on selvittää miten yritykset käyttävät ketteriä menetelmiä liiketoiminnassaan Suomessa ja Brasiliassa. Miten menetelmiä sovelletaan, millaisia ongelmia he ovat kohdanneet ja miten niitä ratkotaan. Brasiliassa on koettu Scrumin sekä muiden ketterien menetelmien käyttäminen vaikeana erinäisistä syistä. Olisi ollut tärkeää saada tietää syitä ja seuraamuksia tästä sekä verrata niitä Suomen tilanteeseen. Tarkoituksena oli siis verrata Suomen ja Brasilian yritysten toimintatapojen eroavaisuuksia ja pohtia niiden syitä, mutta takaiskusta johtuen käytössä on vain Suomessa olevien yritysten haastattelutulokset.

8.1 Yrityksien valinta

Haastateltavien yritysten valintaan vaikutti niiden erilaiset liiketoiminnan alat, halusin tuloksien tulevan erilaisista taustoista. Myös yritysten keskeinen sijainti vaikutti valintaan. Ensimmäinen yritys oli Contribyte, he kouluttavat muita yrityksiä ketteriin menetelmiin ja auttavat heitä tuotekehityksen parissa, yrityksen internet sivullaan tuodaan selkeästi ja heti esille ketterien menetelmien tärkeys liiketoiminnassaan. Toinen yritys oli SprintIt, joka toimittaa avoimen lähdekoodin toiminnanohjausjärjestelmiä yrityksille. Heillä on käytössä toiminnanohjausjärjestelmien ketterä käyttöönotto, joka kiinnitti huomioni miettiesäni haastateltavia yrityksiä. Seuraavat haastatellut yritykset olivat Espeo ja Ambientia, jotka ovat puhtaasti ohjelmistoyrityksiä. Ohjelmistoyritykset ovat tärkeä osa tutkimusta ja valitsin nämä niiden koon ja sijainnin mukaan.

8.2 Haastattelu

Kokosimme haastattelukysymykset (Liite 1.) yhdessä Utzin ja Cardoson kanssa ja suoritin haastattelut neljässä erilaisessa yrityksessä Suomessa, kaikilta haastateltavilta kysyin pääsääntöisesti samat kysymykset. Haastattelut tapahtuivat yritysten omissa tiloissaan. Haastateltavat henkilöt eivät tienneet etukäteen kysymyksiä, mutta olin kertonut heille opinnäytetyöni idean, jotta he pystyivät valmistautumaan haastatteluun.

Haastattelut nauhoitettiin ääninauhurilla ja jokaisesta haastattelusta on oma äänitallenteensa, joista seuraava kappale on kirjoitettu.

8.3 Tutkimustulokset

Kaikissa haastatelluissa yrityksissä ketterät menetelmät ovat suuressa roolissa heidän liiketoimintaa. Kolme neljästä haastatellusta kertoo Scrumin periaatteita

sovellettavan päivittäin. Myös Kanban-menetelmää käytetään kolmessa yrityksessä, SAFe puolestaan on noussut suosituksi menetelmäksi suurissa projekteissa Contribytellä. Toiminnanohjausjärjestelmien toimittamiseen Sprintlt on todennut vesiputousmallin olevan paras menetelmä, muilla osa-alueilla he soveltavat Kanbania. Vain Ambientialla käytetään pelkästään Scrumia, muut soveltavat sen periaatteita paljonkin erilaisissa projekteissa, mutta eivät välttämättä aivan oppikirjan mukaan, lainatakseni yhtä haastateltavista; Scrumin periaatteita ei käsitteellisesti niin noudateta, mutta ajatustasolla monet asiat menevät sen mukaan. Kolme neljästä yrityksestä toivat tämän asian esiin.

Kaikki haastateltavat kertoivat, että projektien tyypit vaihtelevat paljon siitä millainen asiakas on kyseessä. Yhden yrityksen asiakkaat koostuvat pääosin start-up yrityksistä. Start-up yritys on siis nuori kasvuhakuinen yritys, jossa yleensä työskentelee vain muutamia ihmisiä. Heidän kanssa työskennellessä on erityisen tärkeää, että on joustavuutta, läpinäkyvyyttä, luottamusta ja ketteryyttä. On yleistä, että start-up yrityksissä on vain toimitusjohtaja ja hänen idea, joten virheille ei ole tilaa. Toisessa ohjelmistoyrityksessä asiakkaat ovat isompia yrityksiä, kuten Suomen valtio, kirkko sekä muut isot julkiset ja yksityiset yritykset. Isojen tahojen kanssa työskennellessä korostuu sopimuksien teko sekä neuvotteluissa sovittujen asioiden noudattaminen. Ambientialla arvioidaan projektin budjetti ja tehdään työtä sitä vasten, eli on erityisen tärkeää saada mahdollisimman tarkka arvio siitä. Sprintlt:n asiakkaat ovat keskittyneet kaupanalalle, johtuen suurimmaksi osaksi siitä, että heidän toimittama ohjelmisto on jotakuinkin suoraan sopiva monikanavaiseen kaupankäyntiin. Contribyte auttaa erilaisia yrityksiä tuotekehityksessä ja he suosivat pidempiä valmennusjaksoja, haastateltava totesi kolmesta neljään kuukautta kestävästä jaksosta olevan tavanomainen pituus, jossa asiakkaat alkavat oppia ja ymmärtää menetelmien periaatteet ja voivat alkaa soveltaa niitä liiketoiminnassaan.

Myös tiimit vaihtelevat paljon asiakkaan mukaan, joskus tiimi voi olla vain yksi työntekijä, jolloin hän on usein samalla myös projektipäällikkö. Kolmessa neljästä yrityksessä tiimin tavanomainen koko on kahdesta neljään työntekijää, mutta suurimmissa projekteissa työntekijöiden määrä voi olla jopa 15. Contribytellä tiimit ovat yleensä elävä verkosto, jossa roolit ja vastuualueet saattavat vaihdella projektin aikana. Kaikki haastateltavat painottivat tiimien olevan hyvin itseohjautuvia ja usein itse yhteydessä asiakkaaseen. Kaikki totesivat myös olevan tavallista, että projektien taustalla toimii projektipäällikkö, joka varmistaa asiakaskontaktin ja pitää huolen budjetista. Vain Ambientialla on aina scrum-mestari sekä usein myös lisäksi projektipäällikkö taustalla.

Yhdessä yrityksistä pidetään Scrumin seremoniallisuutta haasteena. Sitä, että koko tiimin pitää olla läsnä suunnittelussa ja hienosäädössä pidetään aikaa vievänä ja itse tuotannon tekemisestä pois. Toisen yrityksen haastateltava nosti esille suunnittelun kahdeksi viikoksi etukäteen tuovan ongelmia, sekä sen että ylläpitoon ja muutospyyntöihin on reagoitava mahdollisimman nopeasti. He joutuvat päättämään varaavatko palasen pyrähdyksestä ylläpidollisiin asioihin vai joutuvatko rikkomaan pyrähdystä hoitaakseen esille tulevat asiat. Toimin-

nanohjausjärjestelmien käyttöönotossa suurin haaste on länsimainen sopimuskäytäntö, se että on totuttu sopimaan kokonaisuuksista ja monet olettavat kaiken toimitettavan tiukkana pakettina. Ketteryuden johdosta kuitenkin jotkin asiat saattavat muuttua, esimerkiksi sisältö ja hinta. Puolet yrityksistä kertoi aikatauluttamisen sekä resurssien käytön tuovan haasteita. Joskus tulee tilanteita, jolloin tietty työntekijä on varattuna jossain ja samanaikaisesti hänen tietotaitoaan tarvittaisiin toisaalla, tällöin aikatauluttamisen ongelma nousee esille. Resurssien käytöstä ongelmaksi todettiin budjetin ylitys tai tullut alihankinnan tarve. Näissä tapauksissa on tavallista neuvotella yrityksen johtoportaan kanssa. Yksi haastateltava totesi ketterien menetelmien suurimman ongelman olevan se, ettei se sovi normaalin yrityksen päätöksentekoon juurikin siksi, että tietyillä osa-alueilla asiat saattavat muuttua.

Yleisenä ongelmana nousi myös vahvasti esille se, etteivät asiakkaat aina välttämättä tiedä mitä Scrum on, miten se toimii ja millaisiin projekteihin sitä voidaan käyttää.

9 JOHTOPÄÄTÖKSET

Ketterät menetelmät ja eritoten Scrum ovat vakiinnuttaneet paikkansa yritysten projekteissa, siksi jopa muiltakin liiketoiminnan aloilta ollaan kiinnostuneita Scrum – projekteista. Scrumin hyvä maine tuo siis sen hyvin yleisen ongelman, että asiakkaat haluavat käytettävän Scrumia, vaikka sen soveltuvuutta omaan tarpeeseen ei ymmärretä. Näissä tilanteissa on tavallista, että asiakkaalle selvennetään miten Scrum toimii ja mahdollisesti käydään neuvottelut uudestaan, haastateltavat kertoivat, että asiakasta ikään kuin opetetaan. Scrum ei sellaisenaan toimi kaikilla osa-alueilla, mutta on hyvin yleistä, että sen periaatteita sovelletaan muuallakin kuin ohjelmisto-alalla.

Ketteryuden ydinpointti on projektin aikana tullessiin muutoksiin reagointi. Se, että yllättäviin muutospyyntöihin vastataan nopeasti ja ne hoidetaan kattavasti, on tärkeää asiakkaille. Muutoksiin reagointi ja ketteryuden yleisesti tuoma ongelma yritysten päätöksentekoon on se, ettei voida sopia kokonaisuuksista, paketeista tai kiinteistä hinnoista. Asiakkaalle tulee myös selvittää, että projektin aikana siihen voi tulla muutoksia, joko hinnan tai tuotteen puolelta. Onkin erityisen tärkeää hoitaa suunnitelmat ja neuvottelut kunnolla kokoon, jotta projektin aikana ei siihen tulisi suuria yllätyksiä ja muutoksia.

Scrumin seremoniallisuutta pidettiin haasteena yhdessä yrityksessä, sillä sen koettiin hidastavan tuotantoprosessia. Yleisestikin seremoniallisuuksia kyseenalaistetaan, mutta pyritään kuitenkin toimimaan ohjeiden mukaan, jotta Scrum toimisi kuten pitää. Kaikkien projektin työntekijöiden tulee siis olla mukana Scrumin erilaisissa palaverissa, vaikka tuntuisi tehokkaammalta, että vain osa tiimistä osallistuisi.

Monet haastateltavat nostivat esille niin sanottuja normaaleja yritysten haasteita, aikatauluttamisesta sekä resurssien käytöstä kommunikaatio ongelmiin. Nämä ongelmat eivät suoranaisesti johdu ketteristä menetelmistä, mutta heijastavat haasteita myös projekteihin.

10 POHDINTA

Opinnäytetyön aihe on yhä ajankohtainen, vaikka ketterät menetelmät eivät aivan uusi asia ole. Henkilökohtaisen ammatillisen kehityksen kannalta olen erittäin tyytyväinen aiheen valintaan, sillä työpaikkailmoituksia tarkastellessa olen huomannut, että monissa paikoissa toivotaan ketterien menetelmien ymmärtämistä sekä osaamista. Opinnäytetyön teoriaosuudessa on todella runsaasti uutta asiaa, jota hakemalla ja lukemalla olen oppinut hyvin paljon. Niin itse tiedonhaku, kuin myös kirjoitus- ja tekstinkäsittelytaidot ovat parantuneet huomattavasti työn edetessä.

Kuten luvussa 6 on selitetty, tutkimus jäi vajaaksi, sillä yhteistyö ei toiminut. Minusta riippumattomista syistä tutkimuksen haastattelutuloksista jäi puolet saamatta. Puute vaikuttaa oleellisesti tutkimuksen laajuuteen sekä myös alkuperäinen tutkimustavoite jää nyt saavuttamatta. Olisi ollut erityisen kiinnostavaa saada tietää millaisia eroja Suomalaisten ja Brasilialaisten yritysten välillä on ja pohtia erilaisten kulttuurien vaikutusta eroihin. Tällainen takaisku on opettavainen, sillä kun ongelmat alkoivat paljastua, oli tehtävä päätöksiä miten työtä kannattaa jatkaa. Siinä heijastuu myös mahdollisia tulevan työelämän ongelmia ja sitä, miten niihin reagoidaan ja lähdetään etsimään ratkaisuja.

Toteuttamieni haastatteluiden tulokset ovat luotettavia, sillä tapasin haastateltavat kasvotusten ja nauhoitin keskustelut, kirjoitin nauhoitteista haastattelut auki. Toki kyse on haastateltavan henkilön tiedoista ja kokemuksista, joten täytyy pitää mielessä ihmisten erehtyväisyys. He kaikki ovat kuitenkin oman alansa ammattilaisia sekä jo jonkin aikaa olleet työelämässä, joten tietoja voidaan pitää hyvinkin luotettavina.

Jatkaessani työtä eteenpäin pelkästään omilla haastattelutuloksilla harmittelin, etten kerennyt haastattelemaan enempää yrityksiä, sillä luotin siihen, että saan Brasiliasta heidän tulokset. Laajemmalla haastattelu otannalla suomalaisissa yrityksissä olisin voinut muokata opinnäytetyön tutkimuksen tarkoitusta ja tehdä sen esimerkiksi isompien ja start-up yritysten käyttämien menetelmien eroista.

LÄHTEET

- Agile Alliance (2001). Agile Manifesto
<http://agilemanifesto.org/iso/fi/manifesto.html> Haettu 21.2.17
- Agile Alliance (2015). Agile 101
<https://www.agilealliance.org/agile101/> Haettu 21.2.2017
- Auer, A., Auer, L., Heinäsmäki, M., Hölttä, J., Kalliala, E. Laanti, M. Laine, K. Lekman, L. Miinalainen, P. Naski, H. Piiparinen, T. Puhakke, H. Pyhäjärvi, M. Pääkkönen, T. Räisänen, S. Sora, H. Taipale, M. Talvio, J. Tanninen, A. Toikkonen, T. Toivola, T. Toro, K. Valsta, A. Väyrynen, V & von Weissenberg, M. (2013). *Ketterää kehitystä*. Helsinki: Finn Lectura
- Don Wells (2009). Agile Software Development: A gentle introduction.
<http://www.agile-process.org/> Haettu 19.7.2017
- Don Wells (2013). Extreme Programming: A gentle introduction
<http://www.extremeprogramming.org/> Haettu 6.3.2017
- Don Wells (2009). Team Empowerment.
<http://www.agile-process.org/team.html> Haettu 18.7.2017
- Free Scrum Master Training (2017). Key Processes.
<http://freescrumtraining.org/the-key-processes/> Haettu 29.8.2017
- Haikala, I. & Mikkonen, T. (2011). *Ohjelmistotuotannon käytännöt*. 12. painos. Helsinki: Talentum
- Kurkela, R. (n.d). Tilastollinen tiedonkeruu. Tilastokeskus.
<https://www.stat.fi/virsta/tkeruu/> Haettu 31.8.2017
- LeanKit Inc. (2017). What is Kanban?
<https://leankit.com/learn/kanban/what-is-kanban/> Haettu 30.6.2017
- Lehtonen, T., Tuomivaara, S., Rantala, V., Käsälä, M., Mäkilä, T., Jokela, T., Könnölä, K., Kaisti, M., Suomi, S., Isomäki, M. & Ylitolva, M. (2014). *Sulautettujen järjestelmien ketterä käsikirja*. Turku: Painosalama Oy
<http://urn.fi/URN:ISBN:978-951-29-5838-2> Haettu 6.3.2017
- Royce, W. (1970). Managing the development of large software systems
<http://agileconsortium.pbworks.com/w/file/attach/52184636/waterfall.pdf> Haettu 21.2.17
- Scaled Agile, Inc. (2016). Welcome to Scaled Agile Framework
<http://www.scaledagileframework.com/about/> Haettu 9.3.2017

Shalloway, A., Beaver, G. & Trott, J. (2009). *Lean-Agile Software Development – Achieving Enterprise Agility*. Boston: Addison-Wesley

Solinor (n.d). DevOps – Tehostettua ohjelmistokehitystä.
<https://solinor.fi/devops-tehostettua-ohjelmistokehitysta/> Haettu 20.3.2017

Schwaber K. & Sutherland J. (2016). The Scrum Guide
<http://www.scrumguides.org/docs/scrumguide/v2016/2016-Scrum-Guide-Finnish.pdf> Haettu 6.3.2017

Taina, J. (2009) Ohjelmistoprosessit ja ohjelmistojen laatu. Luentomateriaali, Helsingin yliopisto. https://www.cs.helsinki.fi/u/taina/opol/k-2009/pdf/luvut7-9_2.pdf Haettu 2.8.2017

Vaara, J. (2005) Ohjelmistotuotanto. Luentomateriaali, Lappeenrannan teknillinen yliopisto. <http://slideplayer.fi/slide/1936392/> Haettu 29.8.2017

HAASTATTELUKYSYMYKSIÄ

Yritys

1. Millainen on ketterien menetelmien rooli yrityksessä?
2. Mitkä ketterät menetelmät ovat suosituimpia? Mitä te käytätte täällä?
3. Koetteko, että ketterien menetelmien käyttäminen auttaa projektien etenemisessä?
- 4.

Agile valmennus:

5. Miten valitsette sopivimman ketterän menetelmän asiakkaallenne?
6. Millaisia ovat tavanomaiset asiakkaanne?
7. Mitä he yleensä hakevat valmennukselta?
8. Kuinka kauan valmennusjakso yleensä kestää? Miten sen jälkeen jatketaan?

Tiimi

9. Onko erilaisia tiimejä eri työalueilla?
10. Kuinka pitkään tiimi on käyttänyt ketteriä menetelmiä?
11. Vähentävätkö ketterät menetelmät ajankäyttöä?
12. Onko menetelmien käyttäminen parantanut tiimityöskentelyä ja yhteishenkeä tiimissä?
13. Mitä tiimin sisäisiä ongelmia on tullut vastaan ja miten niitä on ratkottu?
14. Kumpi: itseohjautuva vai käskynalainen tiimi? Miksi?

Scrum

15. Miten ymmärrätte Scrumin; mikä sen pääperiaate on ja miksi käytätte sitä?
16. Millaisissa projekteissa käytätte Scrumia?
17. Kuinka paljon teillä käytetään Scrumia?
18. Miten Scrumin periaatteita teillä sovelletaan?
19. Millaisia ongelmia olette kohdanneet? Miten niitä on ratkottu?
20. Onko tiimiltä hukkunut keskittyminen/motivaatio projektiin Scrumin periaatteiden takia?

Projektipäällikkö:

21. Onko teillä jokaisessa projektissa nimitetty projektipäällikkö? Jos ei, miksi?
22. Millainen on hyvä tiiminvetäjä/projektipäällikkö?